*DAAD 2023, Workshop The impact of pandemic years to informatics education: review and next steps*
*Shkodër, Albania*

"Ss. Cyril and Methodius" University in Skopje
**FACULTY OF COMPUTER SCIENCE AND ENGINEERING**

# A comparison of students' achievements in *Software quality and testing* course during the pandemic and after

*Bojana Koteska*

*University Ss. Cyril and Methodius, Faculty of Computer Science and Engineering, Skopje, N. Macedonia*

*e-mail: bojana.koteska@finki.ukim.mk*

# Outline

- Course overview
- Course syllabus
- Organization during the pandemic
- Sample exams tasks
- Students' results comparison

# Course organization

- Mandatory course in 6th semester since 2020
  - 1 English and 3 Macedonian groups
  - study program: Software engineering and information systems
- 12 Lectures – 2 invited from the industry
- 5 homework tasks – practical tasks
- Exams: theory and practical, max. grade 8
- Projects (optional for students with grades 7 and 8)
- Book: Second edition of “Introduction to Software Testing” by Paul Ammann and Jeff Offutt

# Course goals

- Understand the need for software testing
- Different techniques of software testing
- Use the knowledge in practice: test real projects

# Lecture topics

- Why Do We Test Software?
- Model-Driven Test Design
- Agile Testing
- Criteria-Based Test Design
- Input Space Partitioning
- Graph Coverage
- Logic Coverage
- Syntax-based Testing
- Managing the Test Process

# Homework tasks

- Computer-based testing tasks of real software programs using the latest software testing packages and frameworks
  - Unit testing: Junit 5
  - Automation testing
  - Mocking objects
  - Mutation Testing
  - Graph Coverage
  - Input Space Partitioning
  - Logic Coverage

# Sample homework – Input space partitioning +JUnit

For the following function, model the input domain using both the interface-based and the functionality-based approach.

- Is the partitioning of the input parameters such that it ensures that the partitions are disjoint? Why? If not, alter the partitioning to ensure this property is satisfied.
- Is the partitioning of the input parameters such that it ensures that the partitions cover the entire domain? Why? If not, alter the partitioning to ensure this property is satisfied.
- Choose a base test and list all the necessary tests to satisfy the **Base Choice Coverage** (**BCC**) criterion. How many tests did you get?
- Write JUnit tests using the BCC criteria for ISP coverage.

```
/**
 * Method that finds the employees that belong to only one of the
 * teams given as input parameters.
 *
 * @param team1 represents a set of all employee IDs
 *              for the employees that belong to team1
 * @param team2 represents a set of all employee IDs
 *              for the employees that belong to team2
 * @return the IDs of the employees that belong to only one of the teams
 */
public static Set<String> notInBothTeams(Set<String> team1,
                                         Set<String> team2) {
```


# Sample homework - Cypress

1. Access the juice shop app https://juice-shop.herokuapp.com/#/ or install it locally https://github.com/sanyapaskova/juiceshop
2. Create test that will add and delete item
3. Create test that will search item that is not in stock and validate that can’t be added to basket
4. Rework the previous test in Page Object Model

# Practical projects

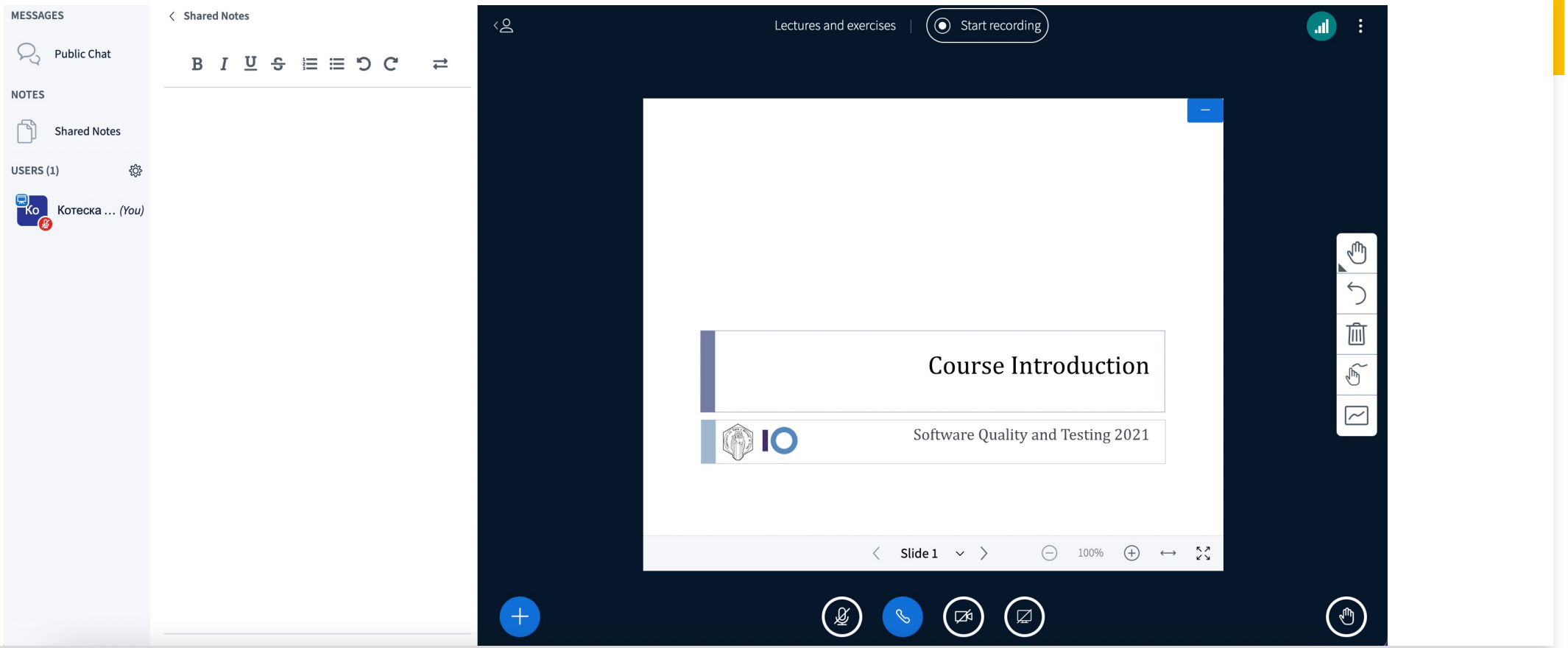
- Teams of 3 students
- Each team has to propose a different real software system for testing
- Multiple techniques/tools have to be covered
- Bonus points – a tool or approach that is not covered in the course




# Ogranization during the pandemics

- Fully remote mode
- Big Blue Button software for lectures – installed on the Moodle platform
- ManyCam for sharing desktop
- DroidCam for connecting phone side camera with ManyCam

# Lectures - BigBlueButton



# Sample exam question – practical task

Question **1**

Not yet answered

Marked out of 10.00

The following code is given.

```
public class VacationPlans {

    public void canGoOnVacation(Person person, VacationDates vacationDates) {
        System.out.println("This program checks if a person can go on a vacation during a particular time.");

        boolean canGo = ((person.moneySaved > person.moneyNeeded) &&
                        (!vacationDates.datesAreWorkingDays || (person.canUseVacationDays == 1)));

        if (canGo) {
            System.out.println("The person " + person.name + " " + person.surname + " can go on a vacation.");
        } else {
            System.out.println("The person " + person.name + " " + person.surname + " can go on a vacation.");
        }
    }
```

**Answer the following questions:**

**1)** Define the predicate that can be derived from the code.

**2)** Find the number of clauses for that predicate.

**3)** Define the clauses for the predicate.

**4)** Choose one of the clauses to be a major clause and answer when that clause defines the predicate.

**5)** Create a truth table and find a test that satisfies CACC when the chosen major clause defines the predicate.

# Sample exam question – practical task

Question 1

Not yet answered

Marked out of 10.00

Consider the following code in which an AOR mutant is given. The given code is a function that calculates the sum of elements of an array.

```
public static int sum (int[] x){
    int s = 0;
    for (int i=x.length-1; i>=0; i--){
        s = s+x[i];
        s = s-x[i]; //AOR mutant
    }
    return s;
}
```

Find a test that does not reach the mutant in the code.

Find a test that weakly kills the mutant.

Find a test that strongly kills the mutant.

Find a test that satisfies reachability, but not infection.

- ✓ Choose...
- half of the elements are positive and the other half negative numbers
- x.length = 0 or x = null
- all elements of the array are zeros or positive numbers
- x.length - is an even number
- x.length = 0
- the sum of all elements of the array is 0
- x[i] =0 , i={0, ... x.length-1}
- x[i] >=0 , i={0, ... x.length-1}
- x[i] >0 , i={0, ... x.length-1}

Edit quiz

Preview

Results

Locally assigned roles

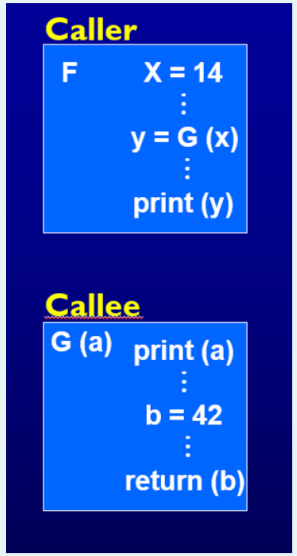
# Sample questions - theory

Question 1

Not yet answered

Marked out of 3.00

Identify the places in the code where last-def will occur when building inter-procedural DU pairs.




Caller: last-def Choose...

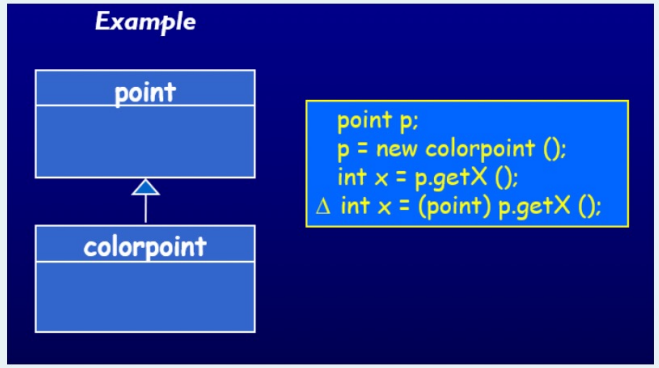
Callee: last-def Choose...

Question 1

Not yet answered

Marked out of 3.00

Which mutation operator is applied in the following case?




Select one:

- OMR –– Overloading Method Contents Replace
- PCI –– Type Cast Operator Insertion
- AMC –– Access Modifier Change
- PPC –– Cast Type Change

# Back to normal

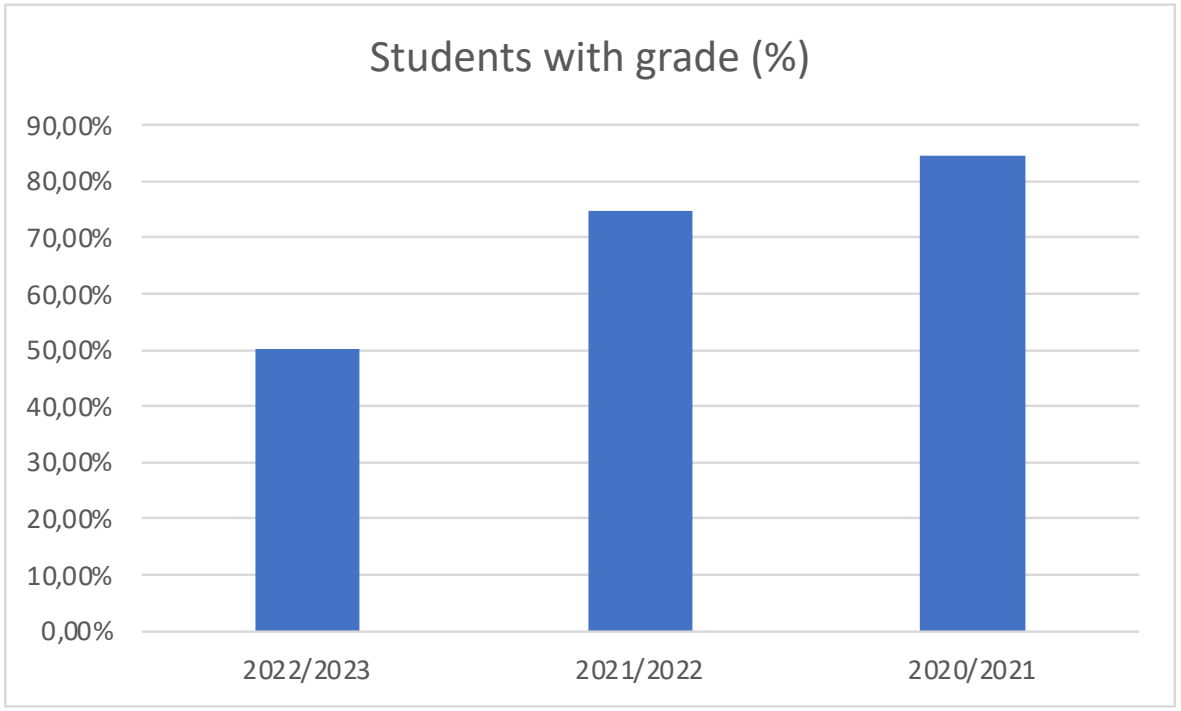
- Lectures face-to-face
- Exams in laboratory
- No cell phones during the exam
- Restricted Internet access, only access to the test

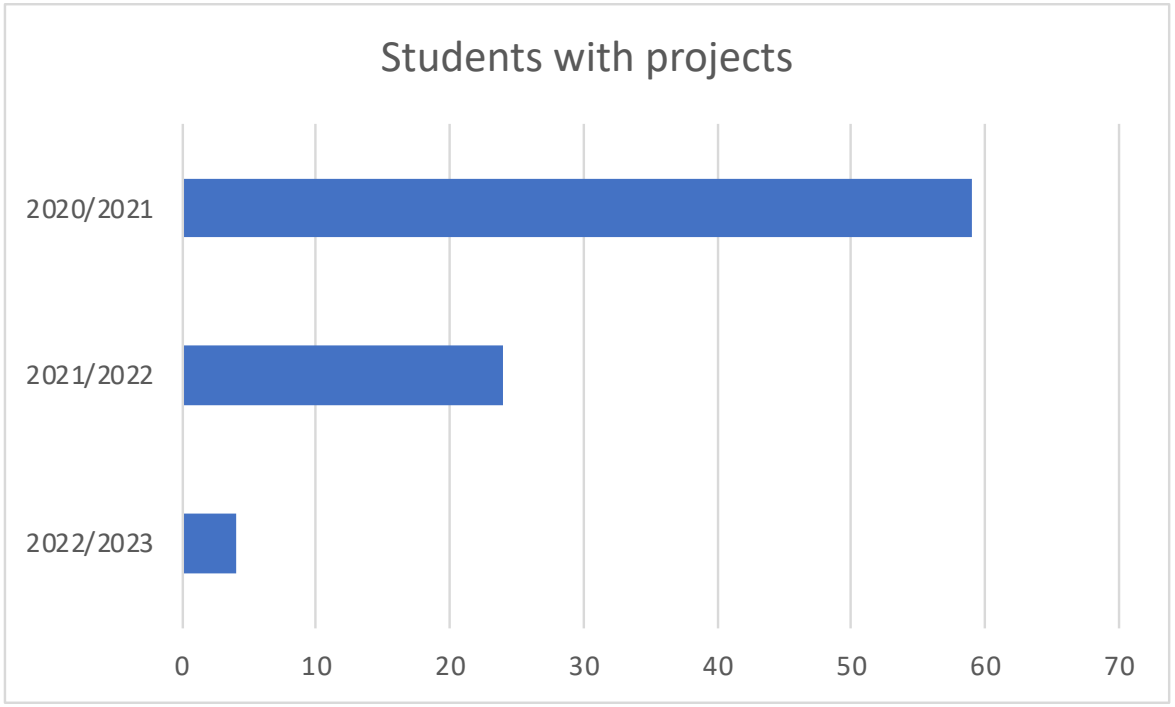
# Student results

| Year | Enrolled Students - (MKD+ENG) | Students with grade (MKD+ENG) | Average Grade (MKD & ENG) | Standard Deviation (MKD & ENG) |
|---|---|---|---|---|
| 2022/2023 * | 273 + 48 | 137 + 23 | 6.89 & 7.00 | 0.78 & 1.13 |
| 2021/2022 | 339 + 45 | 253 + 28 | 6.80 & 7.07 | 0.87 & 1.25 |
| 2020/2021 | 253 + 43 | 214 + 35 | 6.77 & 7.00 | 1.19 & 1.33 |

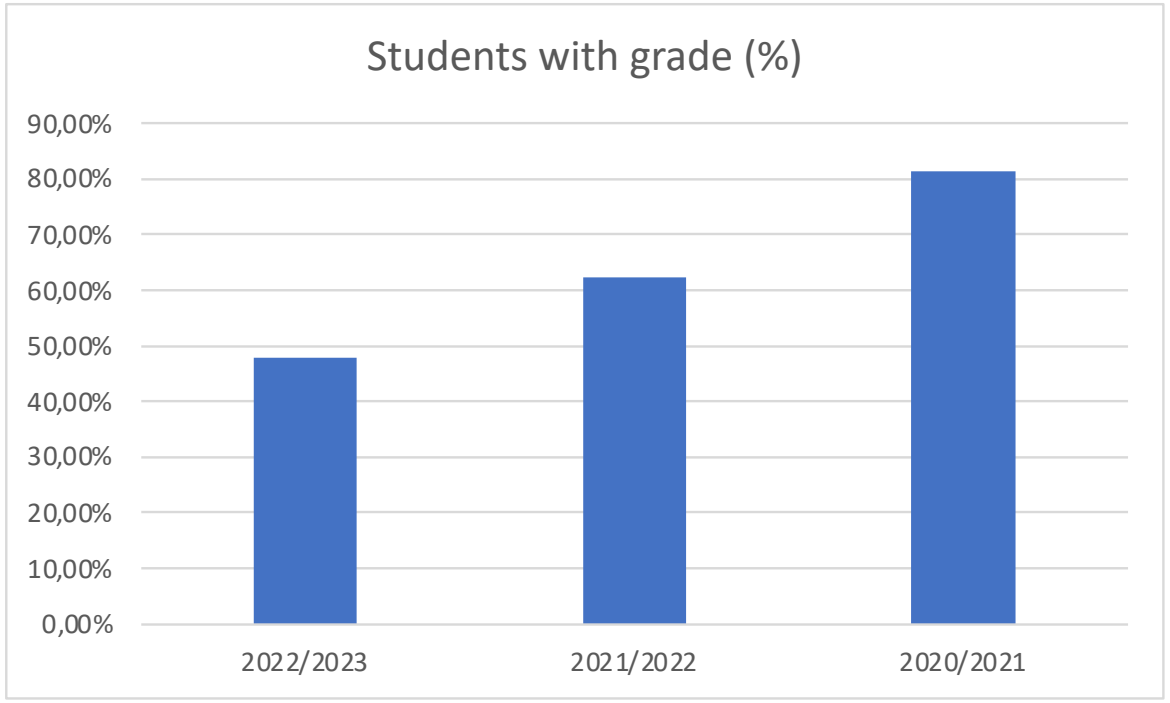
**only in the June exam session*

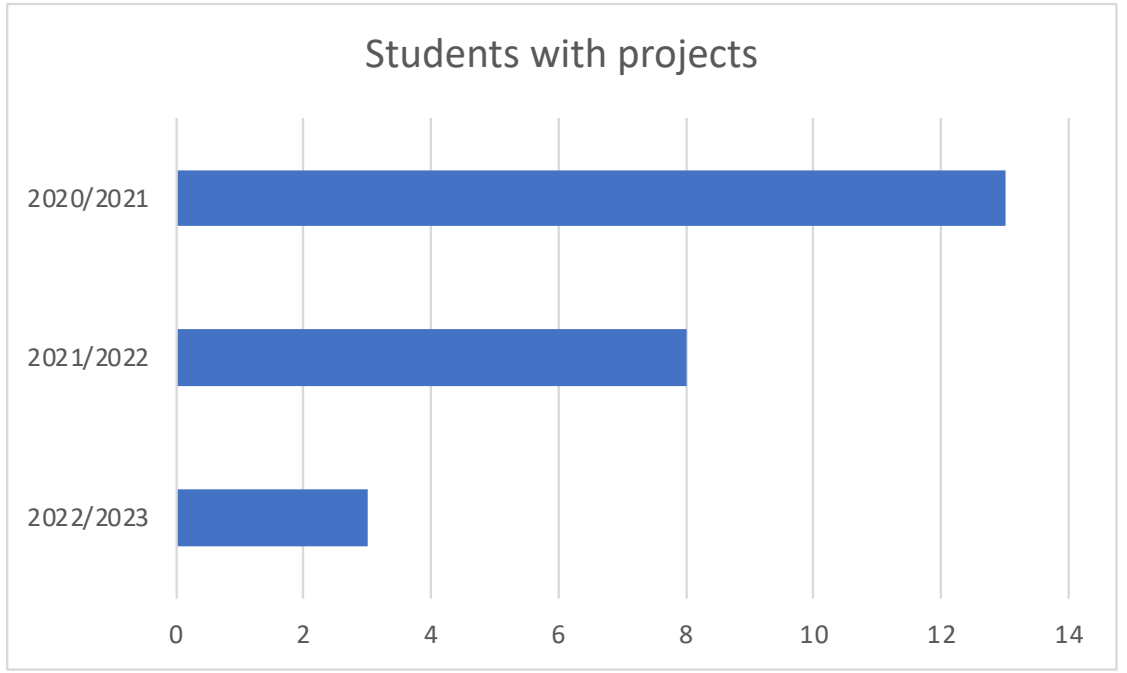
# Result Analysis – MKD groups



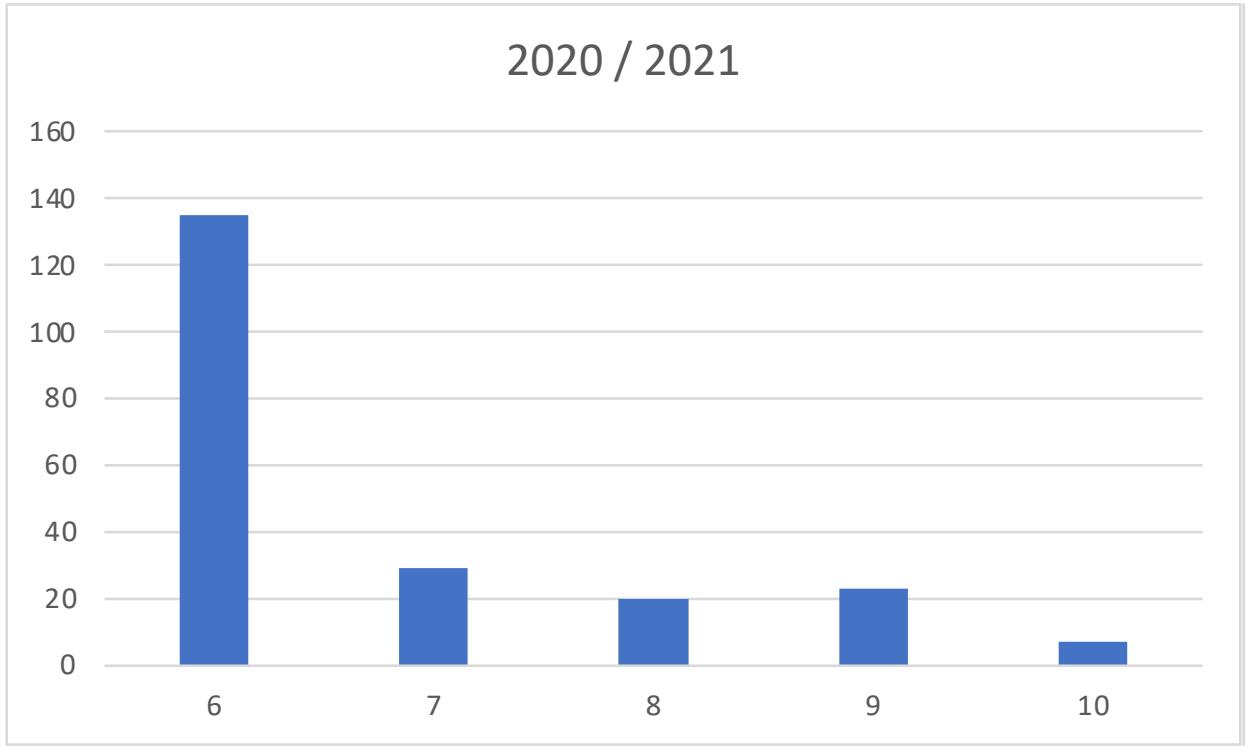



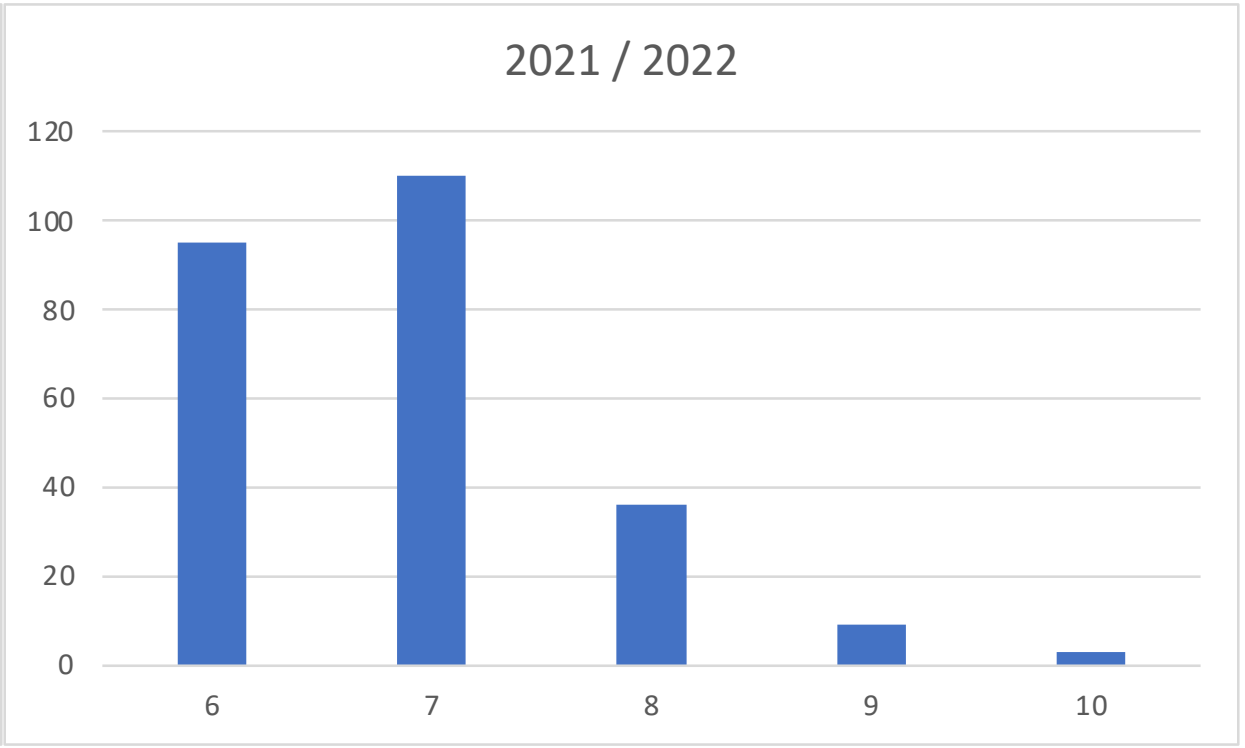
# Result Analysis – ENG groups



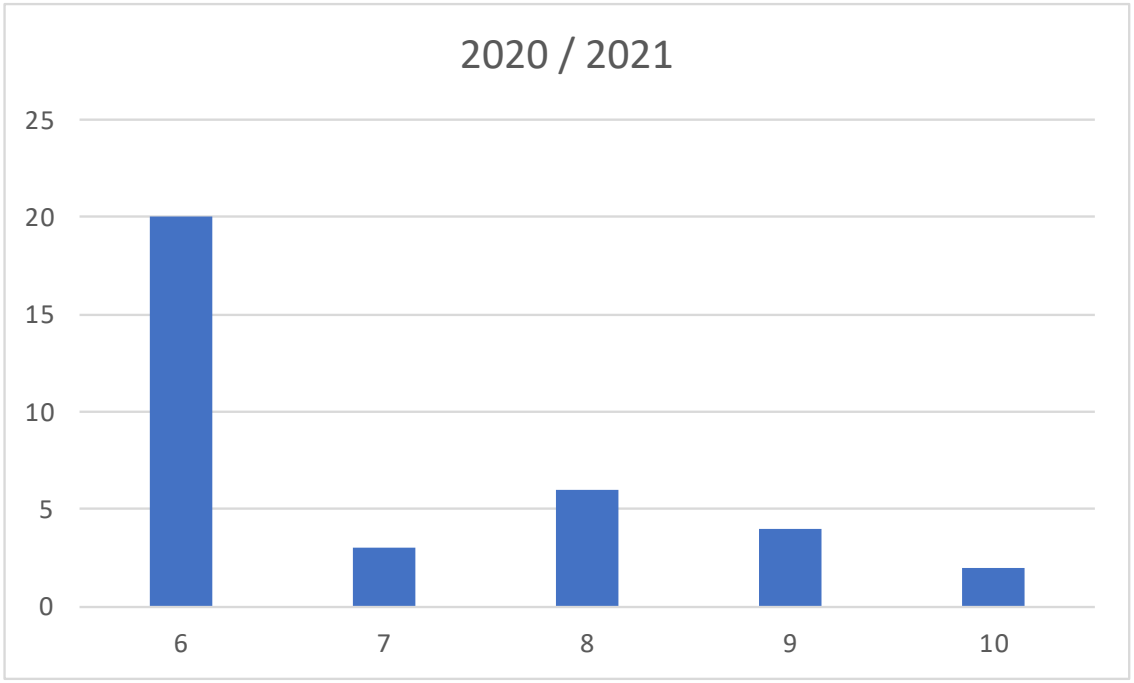



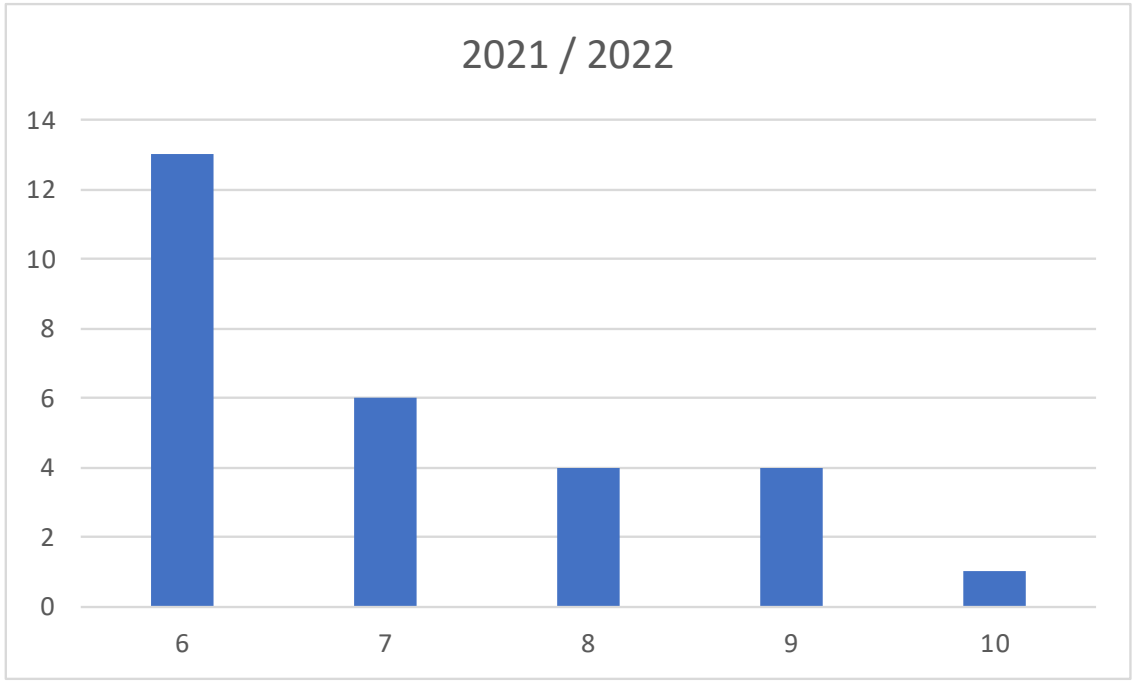
# Grade distributions - MKD



# Grade distributions - ANG



# Conclusions

- More students passed the course during the pandemic
- More students worked on the projects during the pandemics
- Average grades are simillar
- Grade distributions are different

# Acknowledgement


- Special thanks goes to my colleagues prof. Hristina Mihajloska and prof. Gjorji Madjarov for the successful local delivery of the course.
- We really appreciate the motivation of the companies to present their work and to offer internships and employments.
- Thank you for your attention.